





Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

https://en.wikipedia.org/wiki/Git



# **Grundbegriffe git**

- Repository: Projektordner inklusive aller Metadaten (Änderungen, etc.)
- Branch: Abzweigung der Codebasis mit seperaten Änderungen
- Remote: Sachen auf einem online repository
- Commt: eine Sammlung an Änderungen mit einer Nachricht
  - Alle Änderungen am Code werden über commits geteilt
- Stage: Änderungen die für den nächsten Commit vorgemerkt sind
- Tags: Maker an bestimmten commits
- Merge: Zusammenführen von commits verschiedener Quellen
- Subproject: Ein repository was in einem Ordner des repositorys ist
- Blob (binary large object): Dateien die kein Text sind





## git-lfs

- Git nur für Text gedacht, da immer diffs verwendet werden
  - Problem mit großen blobs
- Lösung: Git large file system
  - → Handhabung großen Blobs
  - → Kein Teil von git (standardmäßig)
    - Zusätzliche Installation notwendig
  - → Mehr Infos unter: https://git-lfs.github.com/

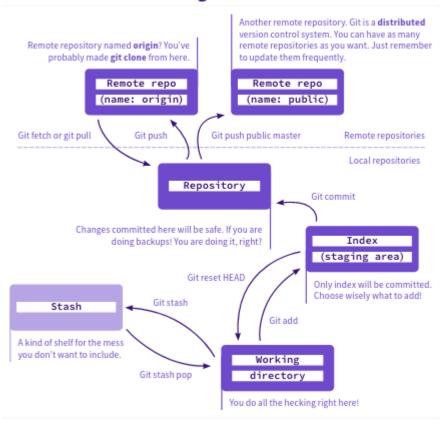
#### MND

Mathematik, Naturwissenschaften und Datenverarbeitung

#### Überblick

TECHNISCHE HOCHSCHULE MITTELHESSEN

#### D The zoo of working areas



UNIVERSITY OF APPLIED SCIENCES Noah Kirschmann Seite



## **Typischer workflow**

- Runterladen des repositories mit: git clone URL
- Ändern der Dateien
- Überblick über geänderte Dateien verschaffen: git status
- Änderungen an einer Datei anzeigen: git diff file
- Änderungen für commit vormerken: git add file1 file2
- Vermerkte Dateien aufzählen: git status
- Änderungen commiten: git commit
- Lokales repository updaten: git pull
- Ggf. merge von remote und local
- Änderungen hochladen: git push



### gitlab

- Issues: todos und gemeldete Probleme eines projekts
- Milestones: Ziele und was für diese erledigt werden muss
- Board: Eine Übersicht über die Issues
- Fork: Eine Abzweigung des Haupt-repository
- Pull-request oder merge-request: Anfrage eine Fork in Main zu mergen
- CI/CD (Continious Intgration/Delivery): Automatisches compilieren, testen und hochladen eines Projektes
- Wiki: Ein wiki des Projektes (Dokumentation u.Ä.)
- Pages: Eine website für Projektdokumentation
- Group: Im Grunde virtuelle Nutzer zum zusammenführen von Projekten



#### Simulator clonen

- Fork vom simulator & libtrainsim erstellen
- Clonen der Fork (Beispiel mit main repo):

git clone https://git.thm.de/bahn-simulator/simulator.git

- Daten herunterladen:
  - Git submodule init.
  - Git submodule update
- Anpassen der Konfiguration:
  - subproject/libtrainsim.wrap öffnen
  - URL durch URL der Fork ersetzten





#### meson

- Buildsystem → Simplere Handhabung von Projekten
  - Cross platform → alle Anweisungen unabhängig vom OS
  - Automatischen Finden von Dependencies
  - Automatisches Erstellen von Compileranweisungen
  - Erstellen von IDE Integration
- Warum meson:
  - Einfache Syntax (python basierend)
  - Hohe Kompatibilität mit anderen Projekten
  - Sehr Anpassungsfähig
  - Schnelle Entwicklung



#### meson Befehle

- meson [command] --help → Hilfe zu command anzeigen
- meson setup build → Konfiguriert den Ordner build als Ziel
- meson configure [options] build → Ändert Projektoptionen
- meson compile -C build → Kompiliert das Projekt im Ordner build
- meson test -C build → Führt die unit tests im Ordner build aus
- Optimierung für CPU: meson configure -Dmarch\_x86\_64=znver1 build
- Dokumentation unter: https://mesonbuild.com/
- Windows build braucht folgende Konfigurationen:

-Db\_lto=false -Ddefault\_library=static

UNIVERSITY OF APPLIED SCIENCES Noah Kirschmann Seite





### kdevelop

- Öffnen des Projektes:
  - Project > import/open project
  - meson.build Datei auswählen
  - Auf finish klicken > Auf okay klicken
- Projekt Kompilieren:
  - Auf das Zahnrad klicken
- Projekt Ausführen/Debuggen:
  - Run > configure launches
    - Simulator > Add > compiled binary
      - Working Directory > simulator
      - Dependencies > build > simulator
      - Fenster schließen
  - Auf execute oder debug klicken